View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by ZENODO

An Application-Aware SDN Controller for Hybrid Optical-Electrical DC Networks

Giada Landi, Marco Capitani
Nextworks
Pisa, Italy
email: {g.landi, m.capitani}@nextworks.it

Domenico Gallico, Matteo Biancani
Interoute S.p.A.
Roma, Italy
email: {domenico.gallico, matteo.biancani}@interoute.com

Kostas Christodoulopoulos
Communication Networks Laboratory of the Computer Engineering and Informatics Department
University of Patras, Greece
email: kchristodou@ceid.upatras.gr

Muzzamil Aziz
Gesellschaft für wissenschaftliche Datenverarbeitung mbH GWDG
Göttingen, Germany
email: muzzamil.aziz@gwdg.de

***Abstract* — The adoption of optical switching technologies in Data Centre Networks (DCNs) offers a solution for high speed traffic and energy efficiency in Data Centre (DC) operational management, enabling an easy scaling of DC infrastructures. Flexible, slotted allocation of optical resources is fundamental to efficiently support the dynamicity of DC traffic. In this context, the NEPHELE project proposes a Time Division Multiple Access approach for optical resource allocation, orchestrated through a Software Defined Networking controller which coordinates the DCN configuration based on real-time cloud application requests.**

Keywords – Software Defined Networking; Optical Data Centre Networks; TDMA.

I. INTRODUCTION

Data Centre (DC) traffic is increasing in volume and dynamicity, bringing new challenges in the design of DC Networks (DCNs) for guaranteeing high capacity, high scalability and energy efficiency together with the flexibility of adapting network configuration based on real-time traffic profiles and application needs. The NEPHELE project [1] proposes a novel DC architecture able to meet these requirements through a disaggregated DC architecture with a flattened DCN infrastructure at the data plane, based on optical switching and slotted resource allocation [2]. At the control plane, the Software Defined Networking (SDN) approach exploits the programmability of the optical data plane and integrates the allocation of the DCN resources in the global management of DC resources, including storage and servers. The open Application Programming Interfaces (APIs) at the SDN controller allow to efficiently coordinate cloud orchestration procedures for service lifecycle management with the provisioning of optical network connections, to transport the intra-DC traffic in isolated, multi-tenant virtual networks.

The interaction between cloud orchestrator and SDN controller is the key to bring application-awareness at the DCN level [3]. Rack-to-rack network connections are established on demand to serve traffic among the DC servers with guaranteed QoS, as driven by real-time application requirements. The automation introduced by the SDN controller reconfigures the DCN to automatically re-adapt the network resource allocation to the current traffic profile. The NEPHELE hybrid electrical-optical DCN data plane enables fine granularity exploiting the slotted resources of the Time Division Multiple Access (TDMA) technologies employed at the data plane. This guarantees the flexibility needed to efficiently accommodate different kinds of cloud applications, as well as high-load DC management traffic for Virtual Machine (VM) transfer and replication.

Previous works have presented NEPHELE architecture [3] and evaluated NEPHELE algorithms through simulation studies [4], while this paper introduces the software prototype of NEPHELE SDN controller for hybrid optical-electrical DCNs and its applicability in DC use cases. The paper is structured as follows. Section II presents use cases for dynamic network allocation in optical DCs, where NEPHELE control plane technologies improve current practices and overcome existing limitations. Section III describes the NEPHELE DC architecture, the hybrid opto-electric infrastructure and the SDN-based network control and cloud orchestration planes. The NEPHELE SDN controller and its software prototype is presented in Section IV. Section V provides conclusions, discussing future research directions in the area of optical DC networks.

II. USE CASES FOR OPTICAL DC NETWORKS

This section presents a set of use cases for DC services with challenges for DCN control, management and cloud orchestration. The use cases are based on current services that may benefit from NEPHELE technologies (Virtual DC), features enhancing existing cloud services (policy-based provisioning of cloud applications with QoS) or solve limitations in DC management (automated disaster recovery and autonomous cloud service upscaling).

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-546-3

A. Virtual Data Centre

The Virtual DC (VDC) use case focuses on the provisioning of highly scalable, customizable virtual instances of DC, delivered as isolated slices dedicated to different tenants but sharing the same physical DC infrastructure. VDC instances are requested by the cloud providers' customers using a web portal. Pre-defined templates define the main features of a VDC instance and the customer can choose different configuration settings (e.g., CPU, RAM, virtual network interfaces) selecting the flavor more suitable to run the desired cloud application. The cloud orchestrator automates the deployment and provisioning of the VDC instance, allocating the virtual resources and delivering the service in real-time to the customer.

The introduction of NEPHELE technologies in the DC architecture is essential to provide network performance guarantees for each VDC instance, exploiting the capabilities of the optical DCN. The integration between cloud orchestrator and SDN controller enables enhanced automation features. For example, scheduled backup or disaster recovery procedures may integrate mechanisms for automated DCN reconfiguration, reserving dedicated resources to the management traffic without affecting the QoS of other running services. Moreover, enhanced monitoring features can be defined on a per-service basis, with monitoring information made available through open APIs for DC administrators and VDC tenants as potential input for Service Level Agreement (SLA) validation tools.

B. Policy-based Provisioning of Cloud Applications

This use-case follows an application centric approach, while the previous one was based on an infrastructure perspective, where the customer required VDC instances replicating most of the features of real DCs. In this use-case, the customer is not interested in the capabilities of the assigned virtual infrastructure, but rather in the application that should run on top of it and in its requirements.

The cloud application platform handles packages that define the application business logic through metadata describing functional and non-functional requirements, like software components, their dependencies, interoperability, auto-scaling rules, etc. Based on these metadata, the cloud orchestrator is responsible to provision the middleware services according to user policies and SLAs, translating application-level requirements in a set of infrastructure-level characteristics and reserving the required virtual resources.

The level of automation in network configuration and resource allocation, enabled through the NEPHELE SDN approach, is an important aspect to reduce the cost of cloud application management. The virtualization of network services, provided by the SDN controller, introduces an abstraction layer that limits the complexity of describing the interconnectivity between middleware services, hiding the details of the DCN infrastructure through intent-based APIs exposed towards the cloud orchestrator.

C. Disaster recovery

The recovery of data and internal states of applications running in the cloud is one of the most critical aspects of cloud services. Cloud providers usually offer several options for management of backups. For example, backup copies of VMs or volumes can be created manually by the customer, following an on-demand approach, in order to save the contents before performing critical operations. Alternatively, they can be triggered periodically as part of the DC management operations. The cloud provider may implement profile-based strategies for storage protection or mirroring, with storage snapshots collected periodically, saved within the same or different DC location. The restoration is triggered by the customer and it is automated through orchestration procedures at the cloud platform.

The traffic generated to move snapshots among servers in the same or in distributed DCs requires large bandwidth, but is delay-tolerant. Its huge load must not impact the traffic generated by the cloud applications running in the cloud, so it would need dedicated connections. In this context the NEPHELE system can efficiently handle the orchestration of the network configuration, reserving intra- or inter-DC connections with the capacity required to enable fast data transfers. These reservations are integrated in the snapshot procedures: they can be performed on-demand in case of customer-driven backups or scheduled with advance reservations and calendar-based mechanisms for periodical snapshots.

D. SLA Monitoring and Automated Upscaling

Elasticity is one of the main benefits of cloud services: cloud applications can scale up and down, on-demand or automatically, based on their load and their compliance with SLAs agreed between service consumer and provider. The NEPHELE system extends this capability to network resources. Open APIs integrate monitoring and dynamic modification of network connections in scaling procedures.

In particular, NEPHELE SDN controller can implement mechanisms to monitor the network performance for each single VDC instance and to evaluate the compliance with the SLA established with the customer. In case of data plane failures or other kinds of SLA breaches related to network performance, the NEPHELE controller can automatically react reconfiguring the virtual infrastructure and establishing alternative paths. These actions may be also triggered in response to failures on computing/storage resources, under the coordination of the cloud orchestrator.

The capability to expose per-tenant network monitoring information through open APIs is an additional feature that can be useful for customers who want to maintain deep control on their virtual infrastructure. For example, these data can be used to detect the need of additional resources in specific VDC instances (e.g., triggering VMs replication to handle variable traffic loads, upgrading of capacity for existing network connections). This option is particularly interesting for VDC tenants who act as service providers and would like to increase and decrease application loads dynamically, reducing costs when resource needs are lower. The cloud orchestrator itself may implement an auto-scaler service, enabling the automated upscaling or downscaling of VDC instances based on customer-driven auto-scaling policies configured at the deployment stage.

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-546-3

III. NEPHELE DATA CENTRE ARCHITECTURE

NEPHELE DC architecture is based on the SDN concept, with decoupling between network data and control plane and their interaction via open interfaces and protocols at the South Bound Interface (SBI) of the SDN controller. This approach allows to customize the network programmability using dedicated SDN applications that interacts with the controller using its North Bound Interfaces (NBI). In DC scenarios, the controller is typically responsible for network virtualization and provisioning of underlying connectivity. On top of that, a cloud management platform orchestrates the whole set of DC resources (computing, storage, networking) interacting with the related controllers and it is responsible to deliver end-to-end cloud services for upper layer applications. NEPHELE architecture is compliant with this trend and proposes a three layer architecture, as follows:

- The DCN data plane employs hybrid optical-electrical technologies to support high capacity and energy efficiency. The TDMA enables high flexibility and fine granularity in resource reservation, to efficiently host different types of traffic flows reducing the overprovisioning.
- The network control framework is based on an SDN controller extended to operate over TDMA-based optical infrastructures. It is responsible for the efficient and flexible configuration of the DCN, in compliance with the requirements of the cloud applications running in the DC.
- The cloud orchestration framework operates the DC infrastructure; it jointly coordinates and orchestrates the resource allocation in the servers and in the DCN, delegating the actual DCN configuration to the network control framework.

A. *Data Plane Architecture*

The data plane architecture of the NEPHELE DCN is represented in Figure 1 and it is based on a flat topology with a two-tier network that can be easily scaled in the east-west direction without increasing traffic latency and congestion. The key of the NEPHELE data plane scalability is given by the definition of *I* parallel planes, each of them including *R* unidirectional rings connecting *P* pods. A pod is made of *I* POD switches and *W* hybrid electrical/optical (Top Of the Rack) TOR switches. Each TOR switch is connected to all the *I* POD switches of the pod, where each port of the TOR switch is connected to a different POD switch.

At the rack level, there are *Z* innovation zones, where an innovation zone is a collection of hosts, storage, memory and other devices. The innovation zones in a rack are connected to a TOR switch with L_E conventional Ethernet links to TOR electrical ports and L_O all-optical links to TOR optical ports. Consequently, each TOR switch has $Z(L_E+L_O)$ ports interconnecting *Z* innovation zones and *I* ports interconnecting to the POD switches.

NEPHELE optical network adopts Wavelength Division Multiplexing (WDM) technology and each of the *R* fiber rings carries WDM unidirectional traffic with *W* wavelengths. The optical links from TOR to POD switches carries a single wavelength at a time, with traffic multiplexed in the time domain using TDMA slots. In the upstream direction wavelength assignment is performed dynamically, per TDMA slot, identifying uniquely the position of the destination TOR switch within the target POD switch. On the other hand, in downstream wavelength assignment is static.

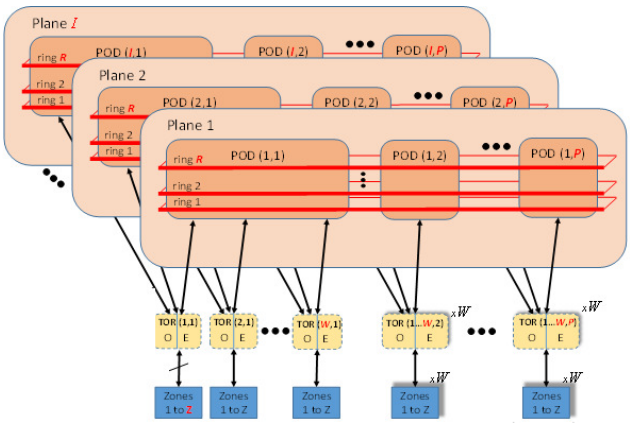

Figure 1. NEPHELE data plane architecture.

B. *SDN-based Control and Orchestration Architecture*

The peculiarity of the NEPHELE network control framework is inherent in its procedures, protocol extensions and algorithms to operate the DCN hybrid data plane applying TDMA concepts in order to obtain a finer granularity and dynamicity in the assignment of network resources. The final objective is the provisioning of intra-DC connections that optimize the usage of the DCN physical infrastructure, while guaranteeing the desired level of QoS for the applications running in the virtual environments.

In NEPHELE, DCN resource allocation is driven from the dynamicity of the traffic demands. These dynamics are captured in a traffic matrix built in real time, based on cloud service requests for new application profiles. Some application awareness is thus transferred from the cloud platform to the network control plane, with extended controller's NBI to enable a more tight cooperation between network controller and cloud orchestrator. This also implies the capability to manage enriched cloud service models at the orchestrator level, with service templates describing network requirements and traffic patterns, as expected by the cloud applications. These parameters constitute the input for the decision engines at the network controller and feed advanced algorithms for application-aware network allocation [4].

The architecture of the NEPHELE control and orchestration infrastructure is depicted in Figure 2. The Cloud Management Platform (e.g., OpenStack) orchestrates the resource of the entire DC and delivers virtual infrastructures to different tenants. All the different devices of the DCN are controlled through a centralized SDN controller which implements the network logic. For scalability reasons, the controller logic can be split across several entities following a hierarchical approach, with child

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-546-3

controllers dedicated to single planes or pods and an upper layer parent controller responsible for coordinating the whole DCN configuration [3].

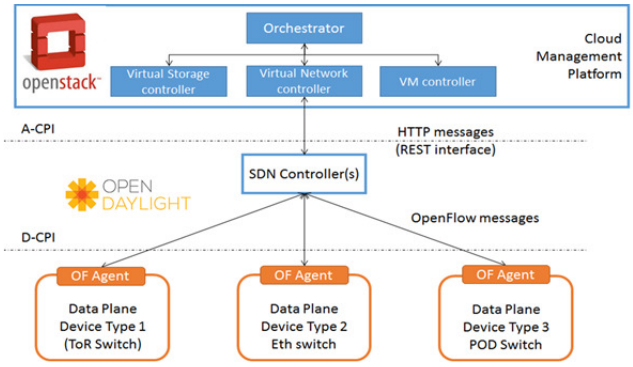

Figure 2. NEPHELE control and orchestration frameworks.

In NEPHELE, the SDN controller prototype has been developed in the OpenDaylight [5] framework. It offers a Representational State Transfer (REST) based interface at its northbound side to receive requests for application traffic profiles, driving the creation of new network connections. Traffic requirements are translated into dynamic network configurations applied to the DCN elements. The interaction with the data plane is based on extended OpenFlow (OF) [6] messages, enhanced to support advertisement, operational configuration and monitoring of optical and hybrid nodes.

IV. NEPHELE SDN CONTROLLER

A. *SDN Controller Functional Architecture*

The NEPHELE SDN controller adopts a twofold strategy for network resource allocation, with real-time reactions for short-term decisions and periodical reconfiguration of the entire DCN for medium-/long-term decisions.

The short-term strategy is applicable to service requests that requires fast activation, to upscaling or downscaling requests of already active services and to react to data plane failures with fast recovery. These cases require high dynamicity and automation of control procedures, so the SDN controller adopts faster "online" algorithms to react quickly to single events, even if leading to suboptimal solutions. The short-term strategy, takes into account single requests instead of the global traffic matrix: it allocates additional resources to serve the new request, given the previous DCN allocation for existing service. This option is well suited for on-demand provisioning and fast recovery of network connections. However, in order to maximize the DCN usage, medium and long-term strategies provide better performance. In this case Application traffic profiles are used as input to build periodically an application-aware traffic matrix. Offline scheduling algorithms elaborates the traffic matrix and re-plan the NEPHELE DCN, computing optimal resource allocation solutions.

Figure 3 shows the functional architecture of the NEPHELE SDN controller, identifying its main components and interactions for short- and medium/long-term resource allocation. The south-bound drivers enable the interaction between SDN controller and data plane; they implement the extended OF protocol for configuring TOR and POD switches and the OVSDB [7] protocol for configuring the OpenVSwitch instances on the servers.

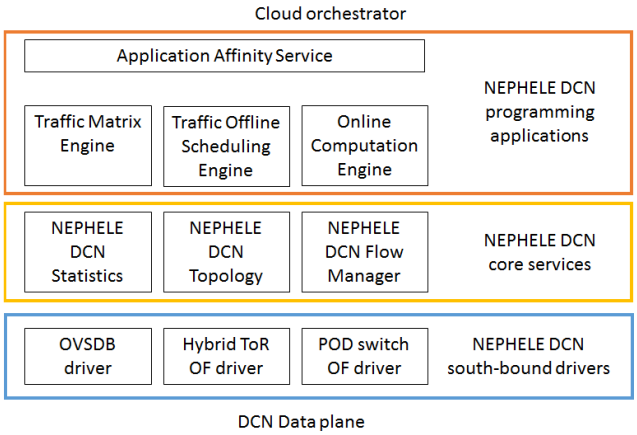

Figure 3. NEPHELE SDN controller: functional architecture.

The core services provide basic functions, abstracting details of the physical resources with unified information models. Core services are invoked by upper layer SDN applications to collect network topology or monitoring information or issue configuration commands through protocol-independent and technology-agnostic interfaces. In compliance with the OpenDaylight architecture, all the services define their interfaces using YANG models [8] and can be invoked by other OpenDaylight components or by external entities via REST APIs.

The NEPHELE network logic is implemented in the DCN programming applications. They employ dedicated algorithms for short- and medium-/long-term decisions at the *Online Computation Engine* and at the *Traffic Offline Scheduling Engine* respectively. The *Application Affinity Service* coordinates the workflows for the DCN allocation, based on applications' traffic profiles. As in core services, all the DCN programming applications expose REST APIs to enable the interaction with the cloud platform.

B. *DCN Configuration Workflows*

This section describes the workflows to allocate network resources in the NEPHELE DCN following the approach based on the periodical reconfiguration of the whole infrastructure, with the optimal allocation solution computed by the offline scheduling engine.

The workflow initiates from the Application Affinity Service, when it receives a request to initiate a network connection for a particular application profile. The details of the requested connections are forwarded to the Traffic Matrix Engine, which updates the current traffic matrix with the new data and returns it. Then, the Application Affinity Service sends the resulting traffic matrix to the Offline Engine, which starts to re-compute the allocation of the network resources for the entire DCN (see Figure 4).

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-546-3

Two strategies for ToR resource allocation are supported. In the *quasi distributed* strategy, the controller takes decisions about slot allocations on output ports, while each ToR decides the source ports to empty. In this case, the traffic matrix is a Kx3 matrix. Each row includes 3 elements *(s,d,t),* where *s* is the source ToR, *d* is the destination ToR and *t* is the number of slots required between *s* and *d*. In the *fully centralized* strategy, all scheduling decisions are taken at the controller. In this case, the *s* element of each row in the traffic matrix represents the source southbound port of the ToR. Details about the engine algorithms and their performance are available in [4].

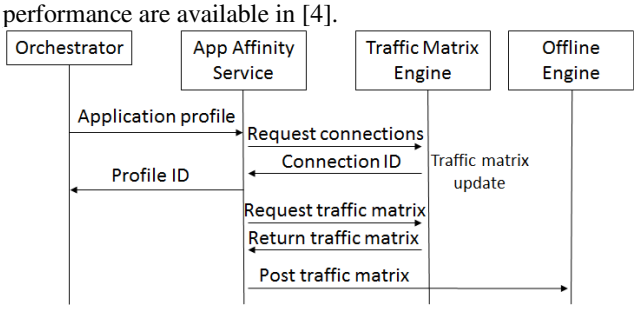

Figure 4. Workflow for creation of a new application profile.

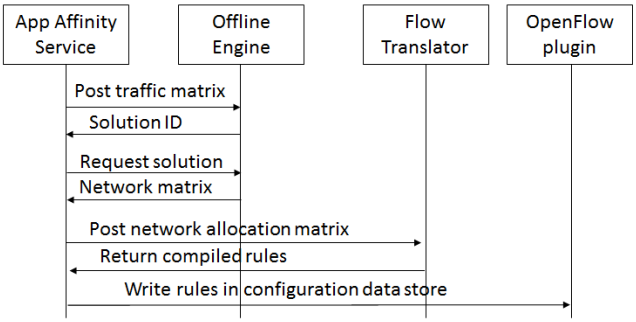

Figure 5. Workflow for updating DCN resource allocation.

Depending on the dimension of the DCN, the engine computation may take time, so the Application Affinity Service requests the network allocation solution with periodical polling queries to the engine until the result is available (see Figure 5). The network allocation provides the details of the time slots and wavelengths to be used for the different connections, taking into account the constraints of continuity along the entire paths (no wavelength or time slot conversion is supported at the data plane). As soon as the computation terminates, the Application Affinity Service forwards the solution to the Flow Manager which de-aggregates the data contained in the matrix and returns the list of flow rules to be installed on the physical devices.

The flow rules defined in NEPHELE extend the traditional rules of the OF protocol. In particular, the flow match structure defines a wavelength and a bitmap of time slots to properly classify the incoming traffic on optical ports, while the same parameters are defined in the flow action structure to specify a cross-connection between two WDM ports for a given set of time slots. The extended YANG model of the OpenDaylight OF plugin is reported in Figure 6, highlighting the extended parameters.

The flow rules resulting from Flow Manager elaboration are then written in the OpenDaylight configuration data store, triggering the procedures at the OF plugin to send the associated Flow Mod messages. At the data plane level, the OF messages are intercepted by device-specific agents, which handle the translation to configuration commands towards the FPGA controlling the hardware.

```
module opendaylight-action-types {
    namespace "urn:opendaylight:action:types";
    prefix action;
    grouping action {
        choice action {
            case output-action-case {
                container output-action {
                    leaf output-node-connector {
                        type inet:uri;
                    }
                    leaf max-length {
                        type uint16;
                    }
                    leaf wavelength {
                        type uint16;
                    }
                    leaf timeslot {
                        type string {
                            pattern '[01]{80}';
                        }
                    }
                }
            }
```

```
module opendaylight-match-types {
    namespace "urn:opendaylight:model:match:types";
    prefix "match";
    grouping match {
        leaf in-port {
            type inv:node-connector-id;
        }
        leaf in-phy-port {
            type inv:node-connector-id;
        }
        leaf wavelength {
            type uint16;
        }
        leaf timeslot {
            type string {
                pattern '[01]{80}';
            }
        }
```

Figure 6. YANG model extensions for encoding of wavelength and time slots in OpenFlow match and action structures.

C. NEPHELE Controller Prototype

The proof-of-concept prototype of the SDN controller developed in the NEPHELE project is based on the OpenDaylight controller, Lithium version, with extended internal components and a set of SDN applications developed from scratch. In particular, for what regards the controller internal modules, the OF OpenDaylight plugin has been enhanced to support the concepts of wavelengths and timeslots in OF rules at the SBI. On the other hand, the software components for the coordination of traffic matrix computation and application-aware resource allocation are

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-546-3

developed as external SDN applications which make use of the controller REST APIs. In particular, the scheduling engine is a standalone application written in C and the algorithm implementation is a translation of MATLAB code converted using MATLAB coder. The other SDN applications are Java applications based on the Spring MVC framework. The controller code is released as open source software and it is available on github [9].

The current implementation provides mechanisms for: (i) accepting requests that specify application requirements in terms of connections between innovation zones with a given reserved bandwidth; (ii) aggregating these requests into a global DCN traffic matrix; (iii) computing a network-wide resource allocation strategy for the current DCN traffic load; (iv) translating the allocation strategy in the set of extended OF rules for the configuration of the NEPHELE data plane; and (v) request the OF plugin to install these rules on the POD and ToR switches.

The NEPHELE controller offers a unified service access point through the REST-based NBI of the Application Affinity Service, enabling its integration with cloud management platforms, like OpenStack. The Northbound API is documented using the Swagger 2.0 tool, which also produces an interactive web-based graphical interface embedded in the application itself.

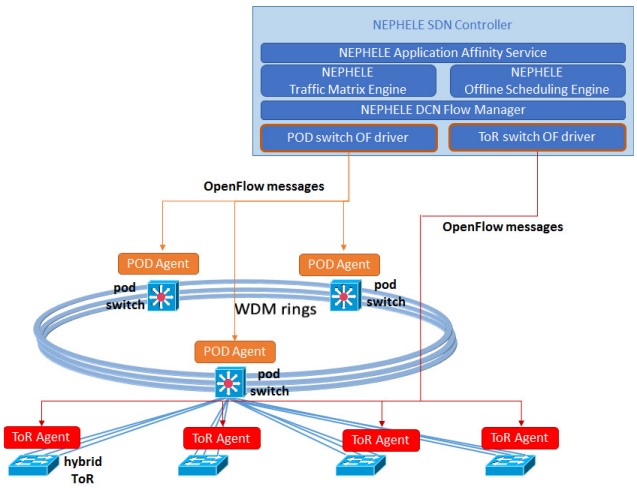

Figure 7. Prototype of NEPHELE controller

Beyond the REST APIs, the OpenDaylight DLUX GUI has been extended to allow requests for provisioning of DCN connections, visualize traffic matrix and flows installed as a result of the Offline Engine computation. The GUI provides a monitoring and diagnostic tool for the DCN administration.

The NEPHELE SDN controller prototype has been demonstrated in a simple environment with a mix of emulated and physical devices at the OFC 2017 conference. The demonstration [10] (Figure 7) shows the entire DCN configuration workflow, from the specification of new application-based connections via GUI, to the elaboration of the optimal resource allocation solution, up to the OF-based interaction with the optical data plane through the OF agents.

V. CONCLUSIONS

This paper has presented an SDN-based control plane for a scalable DCN based on hybrid opto-electrical devices with TDMA technologies, designed in the NEPHELE project. An architecture for the intra-DC infrastructure has been proposed, together with solutions for efficient network resource allocation, based on the global DCN traffic as declared in applications' traffic profiles. Finally, the paper has introduced the proof-of-concept prototype of the NEPHELE controller.

The next steps in the NEPHELE research involve the integration of the SDN controller in a wider environment for the provisioning of inter-DC services. The project will build a hierarchy of controllers, where child controllers will be responsible for resource allocation in single DCNs and in inter-DC network domains (e.g., based on flex-grid optical technologies), while a parent controller will coordinate the end-to-end service provisioning.

ACKNOWLEDGEMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645212 (NEPHELE).

REFERENCES

[1] NEPHELE project web-site: http://www.nepheleproject.eu/ [retrieved: March, 2017].

[2] K. Christodoulopoulos, K. Kontodimas, K. Yiannopoulos, E.Varvarigos, "Bandwidth Allocation in the NEPHELE Hybrid Optical Interconnect", 18th International Conference on Transparent Optical Networks (ICTON), Trento, 2016, pp. 1-4.

[3] M. Aziz et al. "SDN-Enabled Application-Aware Networking for Datacenter Networks", 2016 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Monte Carlo, 2016, pp. 372-375.

[4] K. Christodoulopoulos, K. Kontodimas, A. Siokis, K. Yiannopoulos, E.Varvarigos, "Collisions Free Scheduling in the NEPHELE Hybrid Electrical/Optical Datacenter Interconnect", 2016 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Monte Carlo, 2016, pp. 368-371.

[5] OpenDaylight web page https://www.opendaylight.org/ [retrieved: March, 2017].

[6] Open Networking Foundation, "OpenFlow Switch Specification" version 1.3.1, ONF TS-007, September 2012.

[7] B.Pfaff and B. Davie, "The Open vSwitch Database Management Protocol", IETF RFC 7047, December 2013.

[8] M. Bjorklund, "YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF)", October 2010.

[9] NEPHELE SDN controller code: https://github.com/nextworks-it/oceania-dcn-controller [retrieved: March, 2017].

[10] P. Bakopoulos, "SDN Control Framework with Dynamic Resource Assignment for Slotted Optical Datacenter Networks", Optical Networking and Communication Conference & Exhibition, OFC 2017, Los Angeles, US, March 2017.

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-546-3